Ns G-496

Submitted to Communications of the ACM

A MULTI-USER COMPUTATION FACILITY FOR [redacted] EDUCATION AND RESEARCH*

[redacted]

UNPUBLISHED PRELIMINARY DATA

30 p

Jack B. Dennis

NASA CR 55133

Department of Electrical Engineering and Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts

NASA Grant NsG-496

Synopsis

cite ... all

Present-day computing facilities are limited in their value for scientific research by inability to interact strongly with users. The full power of a research computing instrument should be available at many terminals that give the user the ability to generate, correct and operate any procedure he wishes, either simple or complex. The qualities of continuous accessibility and fast response to simple tasks are crucial in this context. The background of this concept and some of its implications with respect to machine design are discussed in the first part of this paper.

The second part describes the implementation of a small-scale multi-user computer system that permits several users to work independently with the machine, and obtain quite satisfactory response using typewriter communication. The system also provides for the connection of special equipment to be operated by the program of a particular user. Some of the ideas developed for this system will be of value in the design of future large-scale facilities.

AUTHOR

*This work was supported in part by the U.S. Army Signal Corps, the Air Force Office of Scientific Research, and the Office of Naval Research; in part by the National Science Foundation (Grant G-16526); the National Institutes of Health (Grant MH-04737-03), and the National Aeronautics and Space Administration (Grant NsG-496); and was sponsored in part by the Advance Research Projects Agency, Department of Defense, through Office of Naval Research Contract Nonr-4102 (01).

INTRODUCTION

With the maturing of the art of computation, we are now in possession of vastly more powerful calculating instruments in terms of speed and memory capacity. Procedures of great complexity are amenable to mechanization, and simple calculations are accomplished so fast that the start and finish cannot be resolved. It is now apparent, however, that our present means of communicating with automatic computers are not compatible with these greater capabilities: Too much effort must be expended to set up even the simplest calculation for automatic operation. Our techniques for mechanizing complex procedures are antiquated, and do not take full advantage of the internal capabilities of present-day machines.

It is now recognized that improvement of this situation requires the ability of a machine to interact strongly with its environment – meaning physical sources of data and equipment that are being controlled, as well as the human users of the machine. The provision of computers with this ability is crucial to their effectiveness for education and creative research.

At present, available commercial machines still fall short of providing the kind of interaction ability needed for progress in scientific applications. This state of affairs evolved from the demands placed on computer manufacturers by their best customers – government and industry – for whom creative scientific research plays a relatively minor role. Present machine designs are oriented toward administrative and operational data processing.

Thus, it falls to the basic research community – our educational institutions and prominent industrial laboratories – to emphasize the need for a transformation in the quality of computation facilities available for scientific work. The M.I.T. Long Range Computation Study Committee has examined this problem in detail,[2] and has reached this conclusion:

> "The computation needs of education and research can be met best by a large, powerful computer system directly controlled from a number of remote consoles which provide rapid access to the full capabilities of the central machine, and a quick response to simple tasks requested by its users."

[redacted]

The subject of this report is a small-scale computer system designed to provide the qualities of access and response envisioned by the Study Committee. The background and motivation for this study are presented in Part I. Part II gives a functional description of the computer system in its initial form. The performance of this system will serve many research and teaching needs, and will help in the planning of a future large-scale facility.

PART I. BACKGROUND

THE CHARACTER OF ACADEMIC COMPUTER USAGE

To understand the character of computation facility needed in an academic environment, the nature of computer usage in a research oriented university must be examined.

Program Development

Perhaps the most significant observation is that the major portion of academic computer usage is devoted to program development. In education computers are used in teaching the arts of computation--programming languages, numerical methods, and other classes of procedures. Students of these principles learn most effectively by writing, testing, and running their own programs. The computer also serves students as a calculating instrument for term projects and theses in which each task generally calls for the coding of new combinations of familiar and original procedures.

Research in computer technique is concerned with the design and evaluation of algorithms for specific classes of mathematical and information-processing problems, and inherently involves the preparation of suitable programs.

The applications of computation to many broad areas of creative scientific and engineering need programs as tools of research. New tools are continually being fashioned in the form of programs of greater power and with more convenience of use.

Program development requires considerable communication between the human user and the machine: The algorithm must be stated to the machine in an appropriate programming language. The source-language program is then converted into a sequence of machine instructions by a compiling or assembling program. The process of program development would end here, were it not for the difficulty of writing down with complete accuracy the typically rather complex sequence of steps making up a program statement. The more complex the procedure, the greater become the problems of accuracy. During the translation of the source-language program, format inconsistencies and other mistakes frequently show up which require editing of the procedure statement. Once a successful translation of the program has been accomplished, errors in the formulation of the algorithm must be traced and corrected. These steps all require communication with the machine, which is painfully slow with current machine-operating practice: A powerful and costly computer must be continually utilized. Therefore programs are generally run in batches with at least several hours, or as much as several days, required to conduct a brief "conversation" between user and machine consisting of a single statement to the computer followed by the computer's response. The staffs of computation centers compensate for this restricted communication ability by providing overelaborate responses from the machine; the user compensates through elaborations on the statement of operations to be performed. These consequences are wasteful of computer capacity and — perhaps more important — wasteful of human effort.

Experimental Computation

A frequent use of digital computation in research work is modeling the behavior of a physical system. Once a program for performing the simulation has been developed, the researcher generally wants to evaluate its performance under a variety of conditions, or to adjust parameters of the system for optimum performance. In general, convenient access and quick response are required for the power of the computer to be of greatest value to the experimenter.

Conducting Experiments with On-line Computation

In many research applications the computer is used as a sophisticated data-processing device in which the data sources are experiments that are being carried out in research laboratories. Some salient examples are nuclear physics experiments with high-energy particle accelerators, direct observation of the behavior of animal nervous systems, and the evaluation of psychophysical experiments. A number of other possibilities awaits adequate computation facilities for serious exploitation.

At present, it is generally necessary to record data during the experiment, and process it later when computer time is available. Not only does this require expensive equipment or tedious transcription, but the delay in processing and presenting results is frustrating to the experimenter. Frequently, the processed data, if immediately available, would enable the experimenter to change the conditions of the experiment to obtain more significant data. In many cases the computer could even control the conduct of an experiment with more accuracy and precision than the human experimenter, and with more flexibility and sophistication than he could afford the time and effort to build into special apparatus.

In some instances, small general-purpose computers are used to process such data on-line. The character of the processing, however, is quite limited by the capabilities of the equipment being used. Computation facilities are needed which make the capabilities of a powerful processor available for on-line experimental applications.

Unexploited Applications

We have examined the way in which present usage of computation in the university suffers from the inadequate access and response properties of our present computation facilities. As more complex procedures are devised for research studies, this lack will become increasingly apparent. Furthermore, there are many problems for which digital computation would be used and services that a computation facility would provide, except for the difficulty of access. But for this problem, the desk calculator would be obsolete. Services such as commonly used numerical procedures, automatic algebraic manipulation, symbolic integration, the indexing and retrieval of documents, can be mechanized effectively with known techniques. These techniques are useless with our present facilities, for it is too much trouble for the user to phrase his simple question to the computer.

QUALITY OF FACILITY NEEDED

From the nature of academic use of computation, the ideal academic computation facility must possess certain basic qualities. Of greatest importance are the qualities of access and response. The facility should be continously accessible to its users and should react quickly to requests for simple tasks of computation.

Terminal Devices

To obtain these qualities it is necessary to provide appropriate terminal devices for interaction between the user and the machine. The medium of communication should be that which provides the greatest convenience for the user.

In our experience with on-line use of computers, a number of devices have proved particularly useful for communicating with the machine.

1. Typewriter. At present, our techniques for preparing procedures for automatic execution are totally dependent on character strings as the medium of communication. The typewriter is now the cheapest and most convenient device for communicating with a machine in this medium. A typewriter requires only a very small portion of the capabilities of a computer for its operation. Available typewriters are far from ideal for this service, principally because they are overly expensive to connect to a computer and are limited in terms of the character set available to the user.

2. Graphic Display. The cathode-ray tube display is a very flexible instrument for presenting information to a user in graphical form. Although more expensive than a typewriter, a display allows a visual presentation of graphs and diagrams which is easier to grasp and interpret than printed lists of numbers. A display may also be designed to permit rapid presentation of text, so that the combination of keyboard and display can substitute for a typewriter. This has the disadvantage, however, that no permanent record is made of a "conversation" between user and machine. There is a variety of possible levels of sophistication in the design of display systems with a corresponding variation in cost. The kind of display system that is most suitable for general use, and offers a good compromise with expense, is a matter of considerable debate.

3. Light Pen. A light pen is a pen-shaped device that responds when a point within its small field of view is intensified on a cathode-ray display. In conjunction with a display, a light pen gives the user a convenient means of entering two-dimensional relationships among quantities or objects, and a handy way of selecting a particular member of a large set of functions or objects.

4. Buttons and Knobs. Buttons and knobs have been found to be valuable adjuncts to a display/light pen combination. Simply by pressing a button, the user may quickly select the mode in which the light pen is being used, or choose some operation to be performed on information that is being displayed. Knobs have been arranged to give manual control over several parameters of a problem so that their effect can be directly observed in the displayed results.

Form of Information-Processing Capacity

In view of experience with the means of machine-user interaction described above, it is desirable to make these media the primary user terminal devices of a research computation facility. The form to be taken by the processing capacity of the facility is the next question to be studied.

Two major alternatives are apparent for serving a large number of users with limited resources: The processing capacity could be realized as a single central system serving many users simultaneously, or the processing capacity could be split into many relatively small units, each serving one user at a time. The use of many small machines is not a satisfactory solution, as it is possible to provide superior service from a central computer system for much less expense through the techniques of time-sharing.

The size of machine that an individual can afford immediately places a limit on the complexity of procedure that he can mechanize and the amount of data that can be processed. On the other hand, an individual user, although he needs the access and response qualities that we have emphasized, cannot in general effectively utilize the computation capacity of even a small computer. He needs to ponder results of a calculation to determine the most appropriate next step. Furthermore, the amount of processing capability and memory, and the quality of interaction with the machine that a user requires varies greatly in the course of time, and always afford an imprecise match to the capacity of a small machine.

Other disadvantages of the small private computer are readily apparent: It cannot be applied to complex procedures requiring large amounts of main storage. Sophisticated compilers would not be available. Internal program files are not possible without inordinate additional investment in equipment.

Properties Required of a Central Data Processor

These arguments lead to consideration of how a centralized computing system can be arranged to provide a high quality of access to its users. The general layout of such a system is shown in Fig. 1. From our discussion of the nature of needed performance, certain fundamental requirements on the design of the system are evident.

1. It must be possible for a user of the system to write, edit, test, and run an arbitrary program in any available language system, through appropriate use of his terminal or console.

2. The operation of one user's program must not be allowed to affect the computation performed by another user's program, except in the time required for completion.

3. The system must be able to execute simple procedures with convenience and efficiency.

4. The central machine must be capable of providing user-program interaction of a quality suitable for the terminal devices, especially the typewriter and display oscilloscope mentioned above.

To attain the quality of access envisioned here, the capability of the central machine must be such that executing responses to simple queries from all user terminals requires only a fraction of the machine's processing capacity. It is also evident that a large computation facility with the above-mentioned properties is only feasible if the system has a large enough body of users to make the average computation load comparable to the capability of the machine.

REALIZATION OF A TIME-SHARED COMPUTER SYSTEM

The general configuration of a system having these qualities is shown in Fig. 1. A powerful processing and memory element is coupled to many remote consoles for communication with users, and a number of data terminals for connection to experiments. A large auxiliary memory labeled program file in the figure holds all programs so that they may be called up by a user from his console whenever needed. Peripheral equipment permits handling of large printing jobs, loading data from special media, graph plotting, photographing of displays, and other miscellaneous functions.

The central processor of the system must be able to shift attention among independent tasks from many users. For this to be successful, the system must have certain properties: It must be impossible for one program to interfere with correct operation of another program except in the time required to complete a computation. Specifically, a program is prohibited from modifying the contents of storage areas not allocated to it, or operating input or output devices not assigned. Ease of preserving all machine conditions pertinent to the operation of a program is essential for quickly shifting from one program to another.

In terms of storage requirements, the process of shifting can be handled in several ways. One way is to have a main memory large enough to accommodate simultaneously all programs that are active at a given time. In this scheme one program may occupy different locations in the memory on the successive times that it becomes active. Hence means must be provided to allow a program to operate correctly from any area of memory. Alternatively, active programs may be preserved on an auxiliary storage device and brought into the main memory to operate in succession. It appears that in a large time-shared computer system, the best choice will lie somewhere between these extremes.

Available computing equipment is not now oriented toward time-shared operation in the sense with which we are concerned here. Machines have been designed to share processor capability among standard input/output equipment, and therefore include the facility to shift the processor's attention among subroutines associated with each in/out device. No attention, however, is paid to protecting one routine from the actions of another, or to saving all conditions pertinent to a program's operation. Although several manufacturers have recently announced computing systems in which a number of programs may run concurrently with complete independence, no manufacturer has yet acknowledged in equipment design the man-machine interaction capability that is necessary for effective experimental computation.

Since the universities have a need for this transformed character of computation facility, it is appropriate that they take the lead in designing, constructing, and evaluating models of the quality of the system that is needed. This reasoning has stimulated two projects at M.I.T. aimed at demonstrating the feasibility and desirability of a time-shared multi-user computation facility. One study is being pursued by the M.I.T. Computation Center, with an IBM 7090 computer used as the central processor.[1,4] The study described here was made possible by the donation of a PDP-1 computer to the Department of Electrical Engineering, M.I.T., by the Digital Equipment Corporation of Maynard, Massachusetts.

PART II. THE PDP-1 MULTI-USER SYSTEM

The PDP-1 General-Purpose Digital Computer

The PDP-1 digital computer is a small-to-medium scale machine, with binary arithmetic, parallel operation, and a magentic-core memory having a 5-μsec cycle time. The system organization of the PDP is shown in Fig. 2. The arithmetic element consists of the accumulator and memory buffer register, with the in/out register included for shifting, multiplication, and division. The machine performs arithmetic with one's complement binary numbers, Boolian operations on 18-bit words, and a variety of shift operations. The basic memory element contains 4096 words of 18 bits which are directly and indirectly addressable. The memory is expandable to 65,000 words. Random access to the expanded memory is obtained through indirect addressing in an extend mode selected by the program.

Standard input/output equipment for the PDP-1 includes a typewriter through which the user may type information to an operating program, and the program may print data and comments in reply. A photoelectric tape reader operating at 400 characters per second is the normal means of placing programs and data in the machine. A paper tape punch may be used to preserve results, and to prepare machine language forms of programs written in a symbolic source language. A display system permits graphical presentation of results as plots on a cathode-ray tube. A light pen can be used with the display as a versatile graphical input to the machine.

The basic machine includes a sequence break feature designed to allow efficient programming for the various in/out devices. When an input device is ready to transmit a unit of data, it interrupts the operation of a main program and gives control to a service routine that accepts the data, then returns control to the main program. A similar sequence occurs when an output device becomes ready to accept the next unit of data.

Normal Use of the Machine

With the standard machine the process of program development usually proceeds as follows: Soucce language programs are prepared in an assembly language in the form of punched paper tapes by using off-line tape-punching typewriters. The source program tape is then processed by the assembly program which punches an object program in the form of a binary coded paper tape. The assembler also produces a paper tape containing the values assigned to the various symbolic addresses and parameters used in the source program. The process of assembly generally uncovers a number of clerical errors in the coding of the source program. The usermmay decide to edit the source program off-line and repeat the assembly, or he may correct the machine-language object program if the errors were simple and few. In any case, all but the simplest programs are certain to contain errors in their coding which do not show up during assembly, and must be tracked down by running the program and examining intermediate results.

A program known as DDT (for "debugging" programs) has been prepared to help the user in his detective work. The user communicates with DDT through the on-line typewriter. By typing requests to DDT, he may examine and modify the contents of any register of his program or data, or search memory for occurrences of a particular instruction or references to a particular location. Break points may be inserted into the program so that computation stops when a specified step in a program is reached, thereby allowing examination of intermediate results.

An editing program is also available for correcting source language programs through the on-line typewriter. Lines may be inserted, deleted or replaced through simple commands interpreted by the editor.

Multi-User Operation of the PDP-1

The objective in this design of a multi-user system around a PDP-1 computer has been to meet the general goals for an academic computation facility outlined in Part I of this paper in the form of a relatively small installation. In particular, we have tried to retain as far as possible, with available funds, the ability of users to communicate with their programs through typewriter, display, and light pen. The modest size of the system allows us to adopt the philosophy of placing the qualities of interaction ability and response time for simple tasks ahead of attaining maximum efficiency of execution for long programs. We have strived to permit each user of the system to specify and operate any procedure that is feasible for the basic computer. And we have designed the system so that the computation performed for one user cannot interfere with the accurate performance of other users' tasks.

Because of the high expense of display equipment, and the limited capability of the PDP-1 in maintaining many simultaneous displays, it was necessary to emphasize the typewriter as the principal means of communication between user and machine. The display and light pen are incorporated in the system and are assigned to one user at a time.

In our machine, only a single core memory unit of 4096 registers is available, and it was not deemed practical to relocate programs within this memroy. Consequently, only one user's program at a time may occupy core memory, and multi-user operation must be accomplished by running programs in time sequence. With several programs operating, each one is placed in the core memory and run for short time intervals while the other programs are preserved in auxiliary storage. A unique magnetic-drum storage serves this function in the present system. It has 22 fields of 4096 words each. One field can be read into the core memory while the former data in the core memory are simultaneously written on a second field. This "swap" takes place in just 33 msec for a complete field of information. Clearly, the speed of the swapping operation is crucial to the performance of the system, and the above-mentioned time presses the limit imposed by the cycle time of the core memory.

At present, the system includes three consoles and one display, although provision has been made for expansion to seven consoles and two display tubes. In line with our objective of providing for interaction with laboratory experiments, the system includes a facility for attaching experimental equipment to be operated by the system.

Sequence of Program Operation

To supervise the operation of users' programs within the system, a section of core memory is reserved for an executive program. One function of the executive routine is to determine the sequence of operation of users' programs. As it was necessary to keep the executive routine small to make as much memory as possible available to users, a round-robin scheme was adopted. Each user is considered in turn, and his program is boought into memory and run if it has a task to perform. Operation of a program is terminated when a preset time limit is exceeded. The program is then returned to the drum and the next user's pgogram placed in operation. The time limit used is approximately 120 msec. Operation of a program is also ended if it has no further computation to perform, that is, if it is waiting for input/output operations to be completed, has reached a halt instruction or has attempted to perform an illegal instruction.

Handling of Console Conversations

In view of the time-shared mode of operation outlined above, it is not possible for a program to exercise direct control of a console typewriter and operate it at full speed. In typical typing functinns performed by the PDP-1, the machine spends most of the time waiting for the typewriter to complete an operation so that the next character may be transmitted. So that this wasted time may be utilized by other functions of the system, and so that a user's program may run a typewriter continuously at full speed, a set of buffer registers ise included within the executive routine for each console. When a user's program gives a type-out instruction, the executive routine enters the character to be typed in the appropriate buffer table. The executive routine periodically interrogates the console typewriters and transmits characters to them from the buffer tables as they are able to accept them. For typewriter input, the executive routine enters characters in the appropriate buffer table as they are typed, and transmits one character to a user's program each time it issues a type-in instruction. If a type-out instruction is given by a user's program and the buffer table is full, or a type-in instruction is given when the buffer is empty, the program is dismissed until the buffer status changes so that it may continue.

As we have explained, normal use of the typewriter in working with the computer is in conversations conducted by a user in brief phrases with his program. Frequently, a single character typed by the user calls for an action by his program. For example, in using the program-testing routine DDT, typing a backspace causes the contents of the next register in sequence to be printed; typing a slash prints the contents of the register addressed by the instruction last printed. The intimacy of interaction, and flexibility of language syntax provided by the possibility of a program reacting to each character typed by the user is quite significant. Its realization is worth considerable design attention and some additional system expense.

In terms of the multi-user system, the character-by-character concept requires that a user's program be placed in the round-robin queue as soon as any character has been typed into its executive routine buffer.

The Console Control Panel

Providing the user with a console consisting just of a typewriter for communicating with his program deprives him of several convenient features that are available in normal operation of the machine with the on-line typewriter. The standard PDP-1 is equipped with six sense switches that can be set by the user and examined by his program to determine the course of a computation. Also, the console of the machine includes a lever that will start operation of a program at an address specified by a group of switches. The most common use of this lever is to return control to a program-testing routine such as DDT after a program that is being debugged has hung up or entered a long or unending loop. At other times, the start lever may be used to initiate a particular phase of computation by one's own program. It is felt that these features are sufficiently valuable to warrant including them as a part of each console of the multi-user system. A small control panel illustrated in Fig. 3 is associated with each console typewriter. The sense switches on the panel substitute for the standard sense switches while a program assigned to that console is operating. The "call lever" returns control to a program previously specified by the user to the executive routine. Each console panel also includes a "console on" switch that a user turns on to indicate his desire to use the facility. In consequence, an administrative routine is placed at his disposal so that he may request assignment of in/out devices and obtain the services of a program loader, program editor, assembly program or debugger according to his need. Turning the switch off, then on, allows the user to communicate with the administrative routine so that assignments may be changed. Leaving the switch off indicates to the administrator that the user is through so that input/output devices and memory assigned to him become available to other users.

The "stop print" switch on each console panel inhibits printing by the typewriter so that the user may type a new command to his program. The "type" indicator light shows when type-in is permissible, that is, the executive routine buffer is not full. The "run" lamp is turned on while the program assigned to the console is in core memory.

The PDP Display System

The standard cathode-ray display system of the PDP-1 plots one point on a 1024 × 1024 grid for each display instruction executed by a program. The horizontal and vertical coordinates are given by 10-bit numbers in the accumulator and in/out register when the display instruction is given. The programmer creates a visual pattern on the display screen by plotting many individual points in rapid succession. In most cases the coordinates of the points are computed as they are displayed; that is, the pattern to be presented is described by an algorithm for generating the points rather than by a list of the points themselves. Thus the major portion of the machine's computational capacity is required to produce a single display of the flexibility and quality that are possible with the standard PDP-1.

If we are to give all users of the multi-user system access to the display with this quality, only one display presentation can be produced at a time, and the majority of the machine's capability must be made available for this purpose. This is not inconsistent with making enough capacity available to other users so that they may carry on typewriter conversations with satisfactory response. On the other hand, the user of a display presentation generally does not need to observe the image continuously, but will be satisfied if the image can be called into view rapidly when desired. The scheme adopted for the present provides each console with a display lever with which a user may activate a display program. Pressing the lever causes the executive routine to activate display instructions in the user's program and assign a major portion of the system capacity to that program until the display lever is released. Two operators working with typical display programs can readily share the use of a single display oscilloscope with this arrangement.

Program File

In the multi-user system it is important to provide a means of filing source and machine language programs on auxiliary storage for several reasons: An internal program file provides a user with the convenience of calling programs from the file by name through typing a command at a console rather than by handling paper tapes. Also there would be a traffic problem with many users attempting to use one tape reader for entering programs as they are needed. Therefore, the commonly used loading, editing, debugging, and assembly programs, and important subroutines, are kept permanently on one section of the drum. Another area on the drum files programs for users as long as they are operating from a console. A user enters his own programs into the drum file through the paper tape reader at the beginning of a session of usage. Then he may retrieve them from the file as needed by typing a simple command. After editing or assembly, newly formulated programs may be entered in the library. Updated programs may be recorded on paper tape before the user leaves his console. When he is finished, his programs are deleted from the file, so that the space is available to other users. Management of the program file is the responsibility of the administrative program.

It would be desirable if users' programs could be stored indefinitely in the system so that paper tapes would be unnecessary and the read-in procedure dispensed with. Another benefit would be the possibility of locating consoles remote from the computer. Such an arrangement is not possible with the present system because the drum does not have sufficient storage capacity.

Relation to Users' Equipment

A major objective of the PDP multi-user system is to provide means of tying physical experiments into the system so that the machine may process data in real time for presentation to an experimenter, or so the progress of the experiment may be accurately monitored and controlled by a computer program.

To provide for these applications, the computer facility must have the physical means of transmitting information to and accepting information from the experimental environment, and means by which the machine may be notified that data can be accepted or are ready for transmission.

In the present multi-user system, operation of special users' equipment is accomplished through a group of in-out transfer instructions specifically reserved for the purpose.

It is undesirable to assign a particular set of instructions to one user permanently. These instruction codes would then be unavailable to other users as long as the set remained specifically allocated. Also, the set of instructions must be illegal for other users of the facility so that they cannot inadvertently interfere with a production experiment.

The scheme adopted allows each user of special equipment to use the same group of instructions for its control. The equipment activated by the instructions is determined by a set of assignment signals that are controlled by the executive routine. When a program communicating with special equipment is brought into core memory and placed in operation, the assignment signal enabling the special instructions to operate the equipment is turned on. The other assignment signals are not turned on, so that the use of the special instructions will only perform the actions desired by the active program.

Sequence Break System

In normal operation, the sequence break system permits a program to perform computation through the operation of a main program while simultaneously performing several input/output functions through a service routine, with no special provisions necessary in the main program. The sequence break feature does more than provide for simultaneous operation of the in/out devices: It is a convenient means by which an external event may cause a program to switch its mode of operation. For instance, a program executing a lengthy computation may be interrupted by typed information so that its progress may be determined, or a display program may be interrupted by a response from the light pen which gives control to a tracking program. The interrupt feature is thus important for establishing strong user/machine interaction. Therefore the PDP-1 time-sharing system has been designed so that the sequence break feature is available to each user. Of course, the executive routine must interrupt the progress of a user's program to perform its functions of handling buffer operation and the sequencing of programs. This is accomplished by an executive sequence break channel that has priority over users' breaks.

Operating Principles

To implement the multi-user system and achieve the goals outlined above, it was necessary to augment the standard PDP-1 computer with some equipment designed specifically for this purpose. The magnetic drum that can swap information with the magnetic core memory was developed by the manufacturer of the PDP-1 specifically to facilitate time-shared operation.

Other additions were, for the most part, minor but important additions to the logic of the standard machine for the following purposes.

1. To prevent interference among programs running for different users.
2. To allow efficient operation of the executive routine in a small amount of memory.
3. To facilitate conversations between users' programs and consoles.
4. To permit direct operation of external devices by user's programs.

The specifice hardware features that have been added are the following.

1. A set of executive instructions which allows the executive routine to perform its functions.

2. An assignment register that contains a memory bound applicable to a particular user, and codes indicating what in/out devices (console, punch, reader, display, and external device) are assigned to this user. It is loaded by an executive instruction.
3. A two-channel sequence break system. The high-priority channel defines executive mode and establishes operation of the executive routine when a trap or interrupt condition is met (as described below). The low-priority channel is arranged to appear to each user as the one-channel sequence break feature of the standard machine.
4. Control logic for up to seven typewriter consoles.
5. An alarm clock for timing operation of user's programs.
6. Selection logic to permit connection of external apparatus for seven users.

The executive routine consists of two sections as shown in Fig. 4. The common portion remains perpetually in core memory and contains the buffers for characters that are being transmitted to each console, or punch and characters typed in en route to a users' program. The common portion contains the routines that service input/output devices as required, and the program sequencing algorithm. The private block of the executive routine contains routines that provide the input/output facility status information and features specific to a particular user. In general, the private block is different for each user in correspondence with differing needs for services and input/output devices.

The private block is always transferred to and from the drum with the user's program when a program swap occurs.

The executive routine operates with the machine in an executive mode which is entered through a trap or an executive interrupt.

A trap arises from any of a number of specific instructions encountered during operation of a user's program, which include the following.

1. Illegal instructions — The halt instruction, instructions having addresses above the memory bound, instructions referring to in/out devices not assigned to this user, and special instructions reserved for the executive routine.
2. Input/output instructions interpreted by the executive routine — instructions operating the console typewriters, tape punch and reader, and drum instructions.
3. Service request — a special instruction that requests an in/out assignment or a programming feature from the administrator.

When a trap occurs, the executive performs the function called for by the instruction causing the trap, and returns control to the user's program. In the case of illegal instructions, control is not returned to the users' program, but the administrative routine is notified so that it may take appropriate action. The organization of the executive routine is illustrated schematically in Fig. 5.

Executive interrupts arise from conditions external to the user's program which require the attention of the executive routine. These are the following.

1. A console typewriter; the punch, or the tape reader requires service.
2. A console switch is turned on or off, or a call button is operated.
3. The alarm clock counts out.

As indicated in Fig. 5, an interrupt causes any necessary servicing of consoles to be done. Changes in console switches are noted. If the clock has rung, signalling the end of a program's unit of computation, the executive routine searches for another user's program that is ready for execution.

The administrative routine normally resides on the drum and is placed in the round-robin queue when its services are required. A user may request assignment of input/output devices or services provided by the administrative routine either by typing the request at a console, or by executing a special instruction in his program.

Comments on Performance and Suggestions for Improvement

The system described here was demonstrated in May 1963, and made available for general use during the following summer. Figure 6 shows the system in operation. Incorporated in the system is a version of the "debugging" program DDT arranged so that it cannot be altered by any action of a user or his program. As the administrative routine is not yet complete, its functions are performed by commands to DDT.

For the activities of editing source-language programs, performing assemblies, and program "debugging" the time-shared facility has already proved quite valuable. In fact, most users perfer time-shared operation to informal use of the standard machine, because of features provided by the system that are not available otherwise.

One equipment-control problem has already been implemented within time-shared operation — the use of the computer to direct and track a high-gain antenna for radio astronomy experiments. This application requires the execution of a short computation once a minute throughout a run of many hours duration, and thus the advantage of time-shared operation is clear.

Although the systems programming for time-shared operation has not yet reached the form envisioned, it is still possible to foresee those aspects of system performance for which the need for improvement will be most serious.

Probably of most importance is the lack of a sufficiently large program file. Storage space on the drum is limited and is more valuable for other purposes.

It is natural at present to think of a large disc storage as a solution. We have been considering the use of the small, cheap, magnetic tape units that have recently become available as being more in keeping with the informal organization of the facility. With the magnetic tape arrangement a user could carry his personal program file to the computer in the form of a four-inch reel of tape and carry the up-dated file away in his pocket when he leaves.

Communication with the user through the display oscilloscope is far from ideal in the present arrangement, as the full capability of the computer is generally required to maintain an image on the display tube as a sequence of individually plotted points. Thus only one user of the system may view a quality display at a time, and the system can do little in addition except handle simple typewriter communication tasks for other users. Two techniques seem especially appropriate for improving the display performance. One is the reduction of the data rate to the display unit and speed-up of image generation through a pattern generator that can produce characters and straight-line segments from encoded data. The other technique is letting the display unit obtain its data directly from the central memory of the computer through a data channel connection so that the attention of the processor is not required. These methods still allow the use of the light pen to select an object in a display image and interrupt the program so that the identity of the displayed object is immediately known.

With regard to communicating with laboratory equipment, the present system does not permit operation of programs that depend on execution in real time for success. Such programs cannot be run correctly without making the processing unit of the system unavailable to all other users of the system, including the executive routine, for the duration of the real-time run. A resolution of this problem is to build multiaccess computer systems with effectively more than one processing unit so that several program sequences may be executed in parallel. Then one sequence could be assigned to a real-time user for as long as necessary, while the other program sequences serve other users and executive functions.

ACKNOWLEDGEMENT

Many persons have contributed to the success of the multi-user computer facility described here. Discussions of the M. I. T. Long Range Computation Study Committee had a great influence on this project, as did work on a similar system developed at Bolt, Beranek, and Newman by McCarthy, Fredkin, and Boilen.[1] We are indebted to many students for ideas and criticism throughout the course of the work. In particular, Robert Wagner suggested the scheme finally adopted for executive operation of the console typewriters, and an analysis by Thomas Hastings was valuable in estimating its performance. John Yates detailed the special logic for the system[3] and Natalio Kerllenevich assisted in its implementation and check-out. The systems programming for time-shared operation was largely done by Michael Wolfberg and Thomas Eggers. John McKenzie, Ralph Butler and John Connolly are thanked for their technical assistance in assembling and checking the hardware.

BIBLIOGRAPHY

S. Boilen, E. Fredkin, J. C. R. Licklider, and J. McCarthy, "A Time-Sharing Debugging System for a Small Computer." 1963 Spring Joint Computer Conference, AFIPS Conference Proc., Vol. 23, pp. 51-57 (Spartan Books, Inc., Baltimore, Md., 1963).

Report of the Long Range Computation Study Group, Massachusetts Institute of Technology, Cambridge, Mass., April 1961.

John E. Yates, "A Time Sharing System for the PDP-1 Computer." Report AFCRL-62-519, Electronic Systems Laboratory, M. I. T., Cambridge, Mass., June 1962.

The M. I. T. Computation Center, The Compatible Time-Sharing System — A Programmer's Guide. The M. I. T. Press, Cambridge, Mass., 1963.

Figure Captions

Fig. 1. General layout of a large time-shared computer.

Fig. 2. Organization of the PDP-1 computer.

Fig. 3. Console control panel.

Fig. 4. Organization of executive routine in memory.

Fig. 5. Flow diagram of executive routine operation.

Fig. 6. The multi-user system in operation.

Top Denn... Fig 1

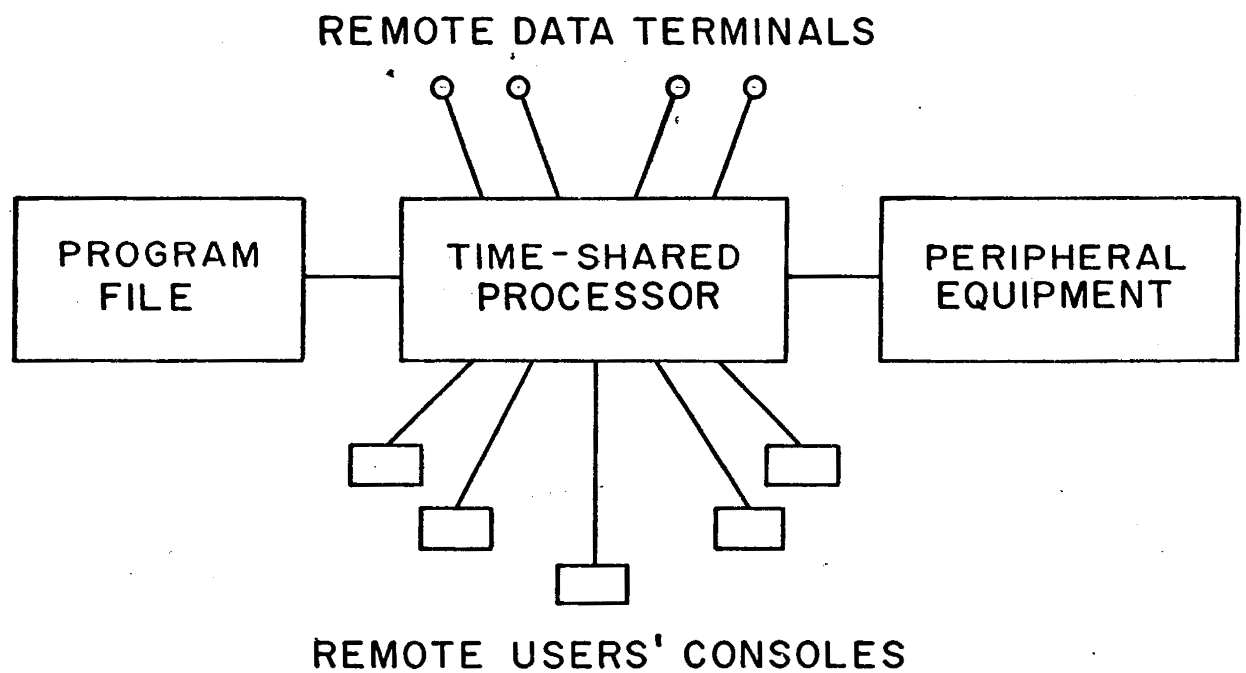

Fig. 1

Dennis Fig. 2

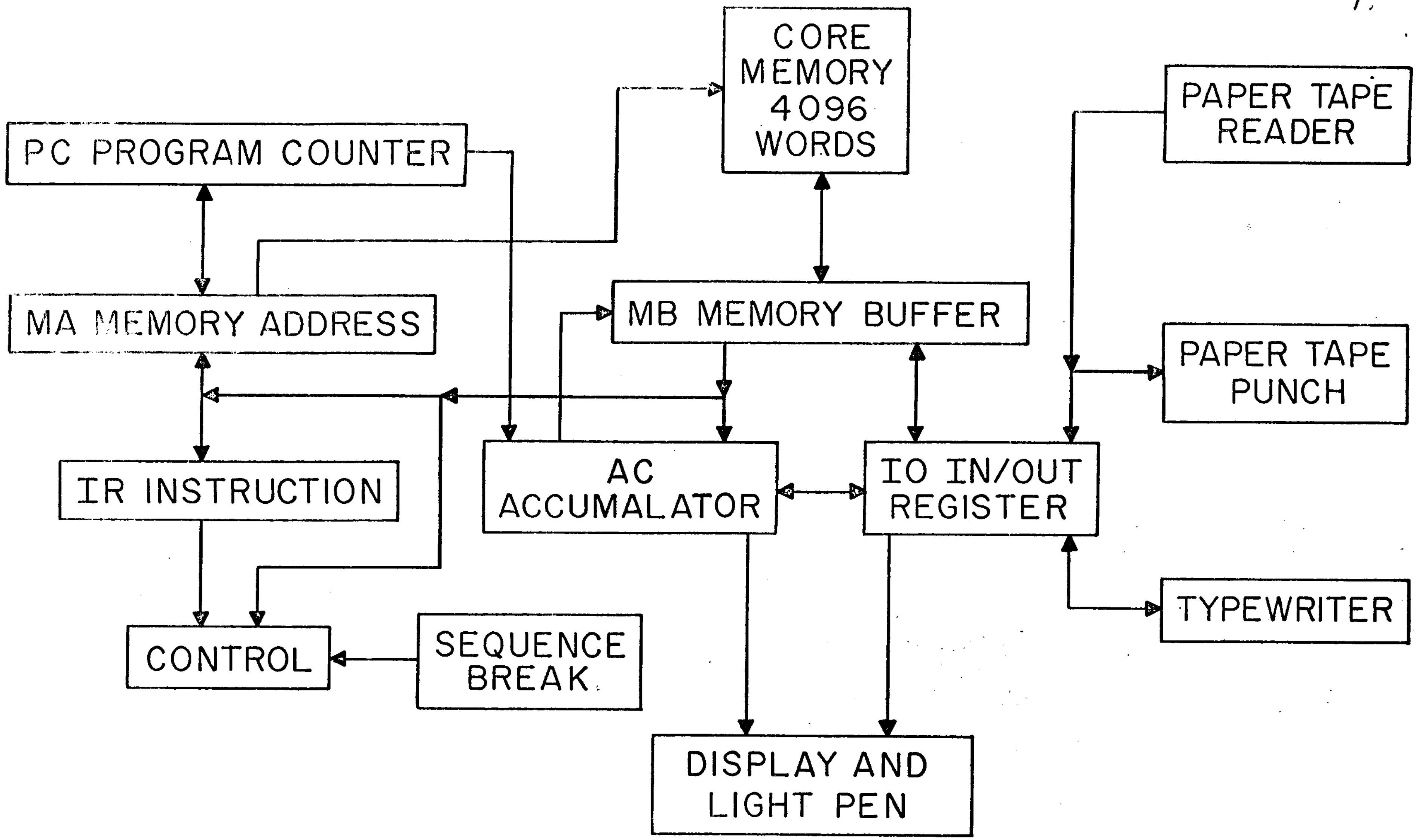

Fig. 2

Dennis Fig 3.

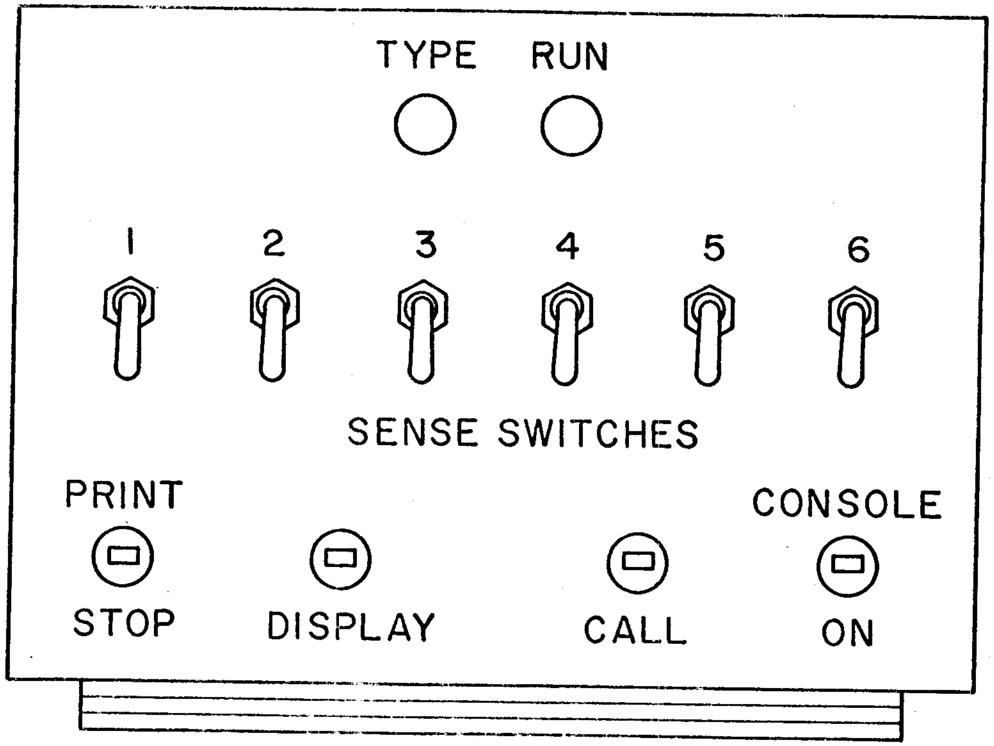

Fig. 3

Top

Dennis Fig. 4

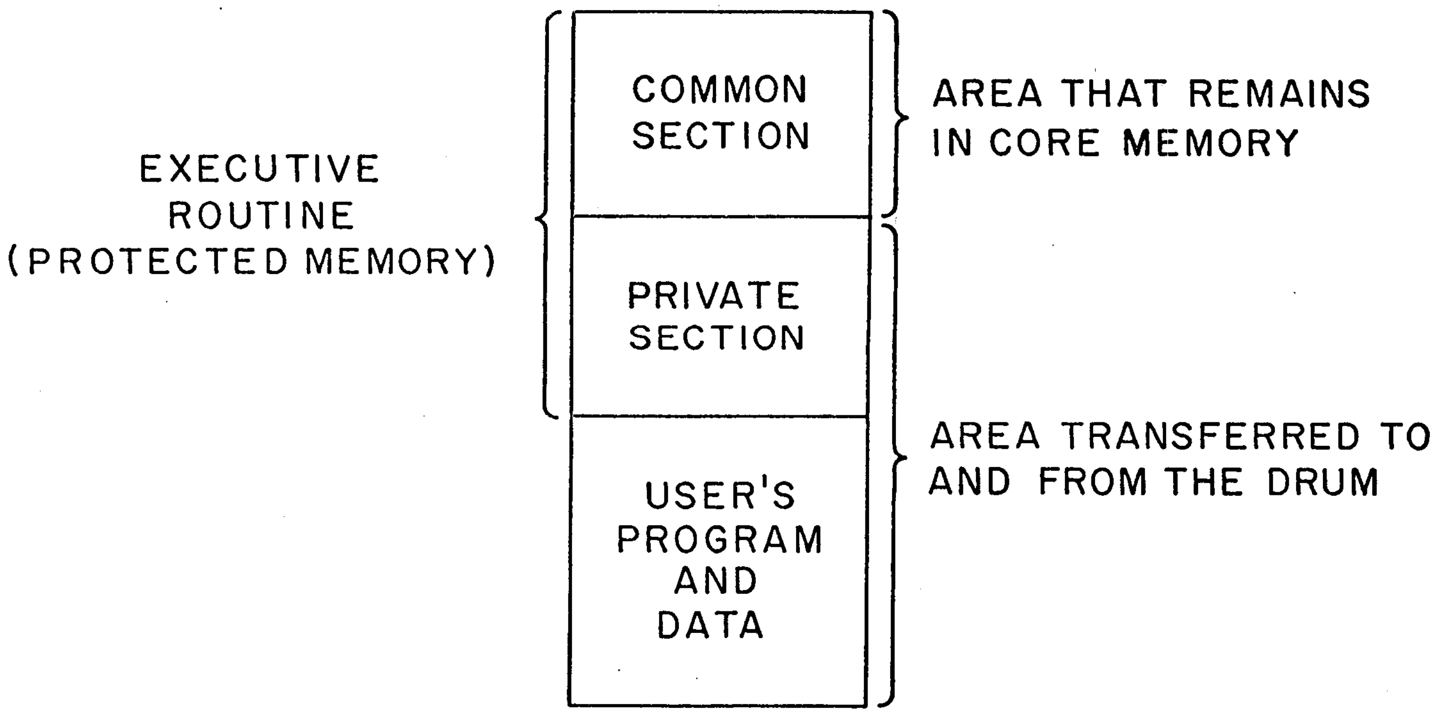

Fig. 4

Top

Dennis Fig. 5

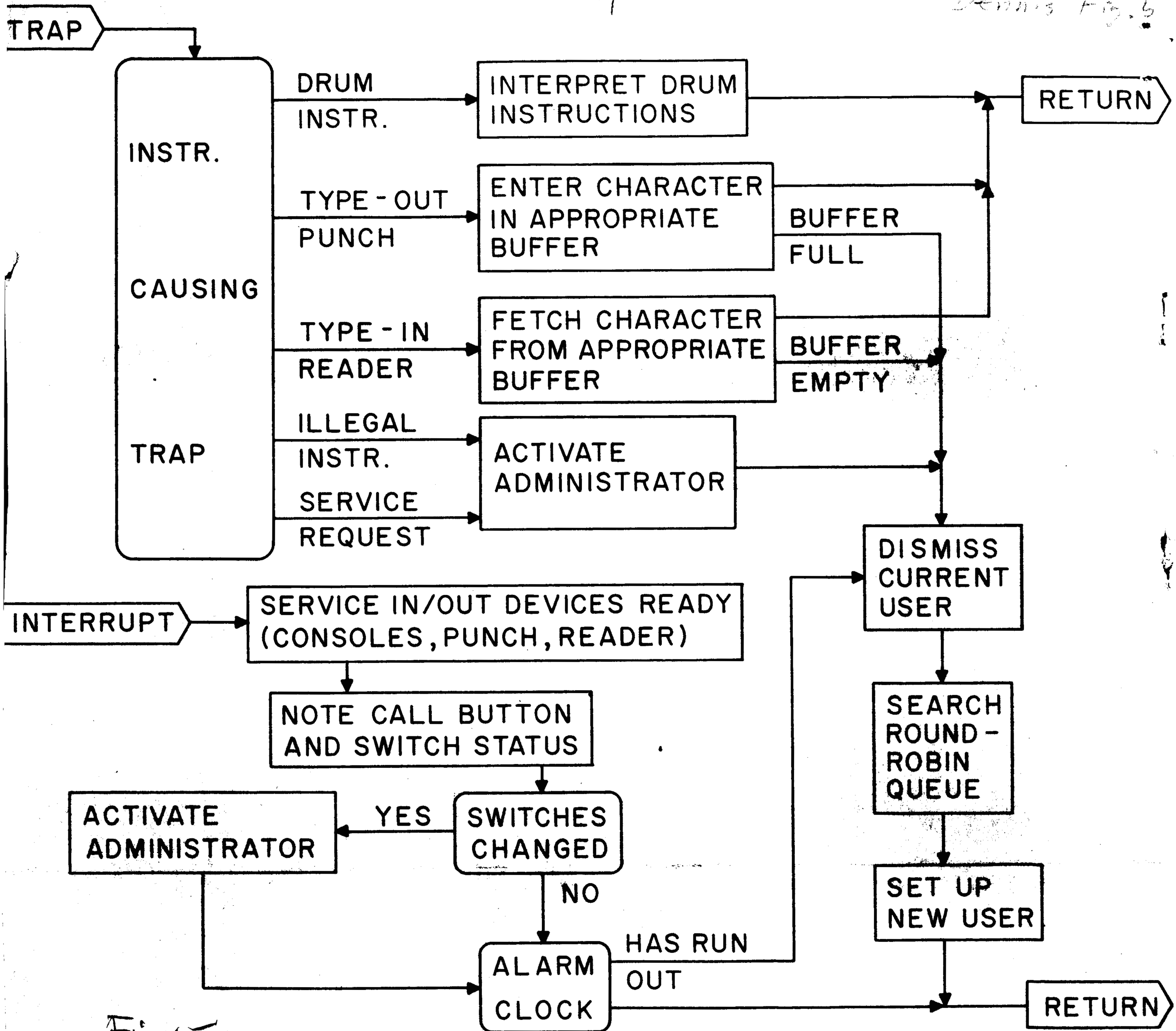

Fig. 5

Top

Dennis
Fig. 6

Fig. 6